

# AUTOMATED LENSING LEARNER - I: AN AUTOMATED STRONG LENSING IDENTIFICATION PIPELINE

CAMILLE AVESTRUZ<sup>1,3,†</sup>, NAN LI<sup>2,4</sup>, MATTHEW LIGHTMAN<sup>5</sup>, THOMAS E. COLLETT<sup>6</sup>, WENTAO LUO<sup>7</sup>

<sup>1</sup>Enrico Fermi Institute, The University of Chicago, Chicago, IL 60637 U.S.A.

<sup>2</sup>Kavli Institute for Cosmological Physics, The University of Chicago, Chicago, IL 60637 U.S.A.

<sup>3</sup>Department of Astronomy & Astrophysics, The University of Chicago, Chicago, IL 60637 U.S.A.;

<sup>4</sup>High Energy Physics Division, Argonne National Laboratory, Lemont, IL 60439;

<sup>5</sup>JPMorgan Chase, Chicago, IL 60603 U.S.A.;

<sup>6</sup>Institute of Cosmology and Gravitation, University of Portsmouth, Burnaby Rd, Portsmouth, PO1 3FX, UK;

<sup>7</sup>Center for Astronomy and Astrophysics, Shanghai Jiaotong University, 800 Dongchuan Road, Shanghai 200240, China;

<sup>†</sup>Provost's Postdoctoral Scholar at the University of Chicago  
avestruz@uchicago.edu

*The Astrophysical Journal, submitted*

## ABSTRACT

Gravitational lensing directly probes the underlying mass distribution of lensing systems, the high redshift universe, and cosmological models. The advent of large scale surveys such as the Large Synoptic Sky Telescope (LSST) and Euclid has prompted a need for automatic and efficient identification of strong lensing systems. We present (1) a strong lensing identification pipeline, and (2) a mock LSST dataset with strong galaxy-galaxy lenses. In this first application, we employ a fast feature extraction method, Histogram of Oriented Gradients (HOG), to capture edge patterns that are characteristic of strong gravitational arcs in galaxy-galaxy strong lensing. We use logistic regression to train a supervised classifier model on the HOG of HST- and LSST-like images. We use the area under the curve (AUC) of a Receiver Operating Characteristic (ROC) curve to assess model performance;  $AUC = 1.0$  is an ideal classifier, and  $AUC = 0.5$  is no better than randomly guessing. Our best performing models on a training set of 10,000 lens containing images and 10,000 non-lens containing images exhibit an AUC of 0.975 for an HST-like sample. However, for one exposure of LSST, our model only reaches an AUC of 0.625. For 10-year mock LSST observations, the AUC improves to 0.809. Model performance appears to continually improve with the size of the training set. Models trained on fewer images perform better in absence of the light from the lens galaxy. However, with larger training data sets, information from the lens galaxy actually improves model performance. Our results demonstrate an efficient and effective method for automatically identifying strong lenses that captures much of the complexity of the arc finding problem. The linear classifier both runs on a personal laptop and can easily scale to large data sets on a computing cluster, all while using existing open source tools.

*Subject headings:* gravitational lensing — methods : numerical — methods: data analysis — methods: statistical — galaxies: elliptical — surveys

## 1. INTRODUCTION

Gravitational lensing occurs when intermediate fluctuations in the matter density field deflect light from background sources (see Kneib & Natarajan 2011, for a review). Strong gravitational lensing can manifest as visible giant arcs magnifying high redshift galaxies (Lynds & Petrosian 1986; Gladsters et al. 2003), multiply imaged quasars (Walsh et al. 1979), multiply imaged galaxies (Sharon et al. 2005), and arclets (Bezecourt et al. 1998). Lensing signatures probe the underlying dark matter distribution of the lens (Warren & Dye 2003), and high redshift galaxy formation (Allam et al. 2007). Strong lenses also provide a geometric test of cosmology via comparison of predicted arc abundances with observed abundances (Kochanek 1996; Chae 2003; Linder 2004), via time-delay between signals from multiply imaged quasars (Suyu et al. 2014, 2016; Bonvin et al. 2017), and via distance ratios in lenses with sources at multiple redshifts (Jullo et al. 2010; Collett et al. 2012; Collett & Auger 2014).

The application of strong gravitational lensing to constrain the mass distribution of strong lenses, such as early-type galaxies (ETGs), necessitates large samples of galaxy-galaxy strong lensing systems. Miralda-Escude & Lehar (1992) first suggested that massive ellipticals would likely be frequent

strong lensing sources in optical surveys. These systems contain a background source galaxy that the lens galaxy deflects into a partial or full arc shaped Einstein ring. The strong lensing signature directly probes the underlying matter. The identification of such systems in upcoming surveys is the first step in constraining the mass-to-light ratio for a large number of objects in this mass range.

Over the last decade, infrastructure for both large scale visual and automated image classification emerged. By nature, the human eye is one of the best discriminators for image classification. Visual arc identification has been effective through the use of citizen science platforms. *SpaceWarps* is an example of citizen science based image classification of strong lensing systems in Canada France Hawaii Telescope Science (CFHTLS) telescope observations (Marshall et al. 2016; More et al. 2016). These platforms are quite successful for a dataset like CFHTLS; here, 3,000 candidate images were identified in eight months, resulting in 89 final candidates. However, future datasets like Euclid (Oguri & Marshall 2010) and LSST expect to find hundreds of thousands galaxy-scale strong lenses (Collett 2015). The volume of upcoming data challenges the scalability of a pure citizen science approach.

Recent efforts have focused on the development of automated methods with performance comparable to or better than humans. *SpaceWarps* is a part of the Zooniverse Project (Marshall et al. 2016), which also includes *Galaxy Zoo*, the citizen science based image classification of galaxy types (Lintott et al. 2008). Galaxy classification is an early example in which machine learning algorithms successfully trained models to classify astronomical images with comparable performance to humans (Dieleman et al. 2015).

Earlier efforts on automated, or “robot”, identification of strong lensing systems have two distinct generalized steps. The first enhances and extracts characteristic features, and the second uses some form of pattern recognition in the features to classify lens and non-lens containing systems. Among others, selected features might include shape parameterization (see Alard 2006; Kubo & Dell’Antonio 2008; Xu et al. 2016; Lee 2017, e.g.), colors in multi-band imaging (Maturi et al. 2014; Gavazzi et al. 2014), light profiles (Brault & Gavazzi 2015), and characteristics of potential lens galaxies (Marshall et al. 2009). Pattern recognition often incorporates cutoffs in selected parameter space distributions (Lenzen et al. 2004; Joseph et al. 2014; Paraficz et al. 2016). A number of these have been publicly distributed, with specific end applications. For example, *ArcFinder* is one such code that finds arcs in groups or cluster scale lens (Seidel & Bartelmann 2007), and *RingFinder* is an analogous tool in searching for multiply imaged quasars (Gavazzi et al. 2014). Codes like these have been complementary to human identification (More et al. 2016).

Pattern recognition methods have transitioned to using “machine learning” algorithms in place of manual cutoffs. With machine learning, we can train a model to separate a dataset into a known set of classes, such as “lensing systems” and “non-lensing systems”. However, many classic machine learning algorithms do not work well in image space, i.e. directly using the raw pixel values of the image, but first require a process of *feature engineering*. *Feature engineering* uses domain expertise to extract variables that are more directly related to the classification task at hand. An example would be the application of an edge detector. The optimal weights and cutoffs for these derived variables that are used to determine the class label of an image are found automatically by the algorithm.

Some more recent works have made use of a subset of machine learning algorithms called neural networks to classify images, either from derived image parameters or directly in image space. In Estrada et al. (2007), authors used derived shape parameters to train neural nets to identify arc candidates. Agnello et al. (2015) used neural networks trained on data from multiband magnitudes, and Bom et al. (2016) used extracted morphological parameters.

The most advanced use of neural networks for strong lensing classification operate directly in image space. (Petrillo et al. 2017) used mock Kilo Degree Survey data to train convolutional neural networks, with a training set size of six million images. Lanusse et al. (2017) used state-of-the-art deep residual neural networks to also work directly in the image space with minimal image pre-processing. A major strength of the Lanusse et al. (2017) implementation is that in comparison to deep convolutional neural networks, deep residual neural networks have been found to be easier to train and perform better (Metcalf et al., in prep).

While there has been a recent surge in the use of deep neural networks applied to image classification problems in astronomy, it is not always easy to scale these techniques to large

data sets, nor are the necessary computational resources and hardware, such as graphical processing units (GPUs), easily accessible to the entire scientific community. We present a supervised classification pipeline built with open source tools. We test the pipeline on mock HST and LSST data, which will also be made publicly available. Our goal is to create a centralized open source pipeline that is general enough for the user to select or add appropriate image processing techniques to augment the features of the lensed image, train and test data with different machine learning techniques, and to select the sequence of methods that provides optimal model performance. Results are reproducible on personal computers, as both the pipeline and the data will be publicly distributed.

This first paper serves as both a presentation of the mock data set and as a thorough case study for our pipeline with one feature extraction and classical machine learning method, respectively, the *histogram of oriented gradients* (HOG) and *logistic regression* (LR). HOG is a fast feature extraction procedure that quantifies edges in images, commonly used to identify humans in security software. LR is a linear classifier, making its scalability relatively straightforward with existing open source tools. This framework is easily extendable to other feature extraction methods and machine learning algorithms for supervised classification.

We show results of the pipeline on mock galaxy-galaxy lens systems observed by HST and LSST as respective examples of classifier performance on optical space- and ground-based observations. We train and test our pipeline on subsamples from 10,000 mock observed strong lensing systems and 10,000 non-strong lensing systems, each centered on a potential lens galaxy.

Our paper is organized as follows. In Section 2 we briefly describe the methods to generate the mock HST and LSST data and our overall image processing and classification pipeline. We present our results in Section 3, and our summary and discussions in Section 4.

## 2. METHODOLOGY

### 2.1. Mock Images

We use mock *Hubble Space Telescope*<sup>1</sup> (HST) and *Large Synoptic Sky Telescope*<sup>2</sup> (LSST) images generated with *PICS* (Pipeline for Images of Cosmological Strong lensing) (Li et al. 2016) and *LensPop* (Collett 2015) for training and testing.

There are 10,000 lens containing mock observations and 10,000 non-lens containing mock observations for running our parameter search. We keep a hold-out set of 1,000 lens containing mock observations and 1,000 non-lens containing mock observations on which we evaluate the final trained model.

Mock observations of lensing systems include the lens galaxy, lensed images of the source galaxy, and galaxies along the line of sight. Mock observations of non-lensing systems include all but the images of a lensed source galaxy. We convolve each of the  $2 \times 10,000$  train/test images and  $2 \times 1,000$  hold-out images to produce three separate sets of mock “observations”. These each have equal numbers of lens and non-lens containing systems: (1) HST-like observations, (2) best single exposure LSST-like observations, and (3) LSST-like observations over the span of ten years. We respectively label these observations as HST, LSST-best, and LSST10.

<sup>1</sup> <https://www.spacetelescope.org/>

<sup>2</sup> <https://www.lsst.org>

### 2.1.1. Modeling the Mass Distribution of Lens Galaxies

To produce simulated lensed galaxies, we first model the mass of lens galaxies as a Singular Isothermal Ellipsoid (SIE). This model is both analytically tractable and is consistent with models of individual lenses and lens statistics on the length scales relevant for strong lensing (Koopmans et al. 2006; Gavazzi et al. 2007; Dye et al. 2008; Li & Chen 2009, e.g.).

The normalized convergence map of the SIE model is defined as:

$$\kappa = \frac{\theta_E}{2} \frac{1}{\sqrt{x_1^2/q + x_2^2}}, \quad (1)$$

where  $\theta_E$  is the Einstein Radius and  $q$  is the axis ratio. The Einstein radius can be calculated from the redshift of the lens, the redshift of the source, and the velocity dispersion of the lens galaxy as follows,

$$\theta_E = 4\pi \left( \frac{\sigma_v}{c} \right)^2 \frac{D_{ls}}{D_s}, \quad (2)$$

Here,  $c$  is the speed of light,  $\sigma_v$  is the velocity dispersion of the lens galaxy,  $D_{ls}$  and  $D_s$  are respectively the angular diameter distances from the source plane to the lens plane and from the source plane to the observer.

To rotate the lenses with random orientation angle  $\phi$ , we adopt the transformation below:

$$\begin{bmatrix} x_1' \\ x_2' \end{bmatrix} = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \quad (3)$$

From Equations 1 - 3, there are five independent parameters from which we can derive the lensing map: velocity dispersion  $\sigma_v$ , axis ratio or ellipticity  $q$ , orientation angle  $\phi$ , redshift of the lens  $z_l$ , and redshift of the source  $z_s$ .

We choose  $\sigma_v$ ,  $q$ , and  $\phi$  randomly (flat prior) from typical ranges of observed galaxies:  $\sigma_v \in [200, 320] \text{ km/s}$ ,  $q \in [0.5, 1.0]$ , and  $\phi \in [0, 360]$ . While a flat prior is not realistic, we use this as a starting point to test our pipeline (see Section 4 for more details on future work). We obtain the redshift of the lens galaxy by matching the velocity dispersion from our simulations to the catalog of elliptical galaxies in the COSMOS survey from Zahid et al. (2015), resulting in lens galaxy redshifts in the range,  $z_l \in [0.2, 0.7]$ .

### 2.1.2. Modeling Images of the Lens Galaxies

We model the light distributions of the lens galaxies with an elliptical Sersic profile,

$$I(R) = I_{\text{eff}} \exp \left\{ -b_n \left[ \left( \frac{R}{R_{\text{eff}}} \right)^{1/n} - 1 \right] \right\} \quad (4)$$

where,  $R = \sqrt{x_1^2/q + x_2^2}$ ,  $R_{\text{eff}}$  is the effective radius in units of arcseconds,  $I_{\text{eff}}$  is the intensity at the effective radius,  $n$  is the index of the Sersic profile, and  $q$  is the ellipticity of the lens galaxy. We perform a similar transformation as Equation 3 to orient the source galaxies and assume that the distribution of light follows that of mass. The ellipticity and orientation are therefore the same as in the SIE model.

We use the COSMOS morphological catalog (Zahid et al. 2015) to match the velocity dispersion with an assigned effective radius, effective luminosity, and index to the light profile. We also assume the light center is on top of the mass center, creating noiseless images of lens galaxies.

We construct galaxies along the line of sight by cutting light-cones from the Hubble Ultra Deep Field<sup>3</sup>. We stack these images with the lens galaxy image and the lensed source galaxy, calibrating the magnitude of all three components.

### 2.1.3. Modeling Images of the Source Galaxies

The background source galaxies come from a set of detailed images of low-redshift bright galaxies ( $z \sim 0.45$ ) that have been extracted from mosaics produced by the CANDELS team (Grogin et al. 2011; Koekemoer et al. 2011) and selected from the CANDELS UDS catalog (Galametz et al. 2013). We rescale these background source galaxies, set the source redshift to  $z_s = 2$ , and select source positions near caustics of the lensing system. The selected positions are within a  $2'' \times 2''$  square box centered on the lens galaxy center of mass. We then fix the projected position of the lens galaxy at the center of the field of view for each image.

We perform ray-tracing simulations using the modeled galaxy images and parameters. The lens galaxy, source galaxy, and parameters of the image simulations (i.e.,  $300 \times 300 \text{ pixels}^2$  with  $0.03 \text{ arcsec/pixel}$  for an HST-like image) are inputs to the simulation that produce ideal lensed images with the appropriate resolution.

### 2.1.4. Mock Observing

The final part is to perform the mock observation from the ideal images produced in the previous step. Both our HST- and LSST-like observations are monochromatic, respectively using the F606W and the g-band. We stack the lens galaxy, source image, and galaxies along the line of sight. While we adjust the source size and magnitude to correspond to the change  $D_s$  when placed at  $z_s$ , we do not adjust colors. The distribution of lens properties is not realistic with a constant  $z_s$ , but provides a sufficient start in covering the feature space to test supervised classification methods. Incorporating the lens, source image, line of sight galaxies, and noise are particularly crucial in methods that use edge features.

For HST-like observations, we do the following for each component. The component that mimics along the line of sight galaxies is a cutout from the HUDF. Stacking the cutout results in an image where noise and the point spread function (PSF) for HST is in the field of view. The lens galaxies have been convolved with the HST PSF, but their angular extent is significantly larger than the PSF of  $\sim 0.03''$ . Convolution of the lens galaxy component will not noticeably alter its appearance. We do not convolve the the source image with the PSF in the mock HST images to best mimic any clumpiness that might be present in an HST arc; the source images are from a ray traced CANDELS galaxy observed with HST seeing, which does not capture the true clumpiness of these sources. We then stack and magnitude calibrate all components to produce the final HST-like images, which are  $300 \times 300 \text{ pix}^2$  with  $0.03 \text{ arcsec per pixel}$ .

For LSST-like observations, we use the *LensPop* software (Collett 2015). We resample images to match the detector pixel scale and convolve the resampled image with a circularly symmetric Gaussian Point Spread Function discretised at the same pixel scale. To generate a noisy realization of the image, we assume a Poisson model based on the sky plus signal, and an additional Gaussian read-out noise. Parameters for these simulations follow Collett (2015) and are based on the LSST observation simulator (Connolly et al. 2010).

<sup>3</sup> [http://www.spacetelescope.org/science/deep\\_fields/](http://www.spacetelescope.org/science/deep_fields/)

To account for variations in seeing and sky-brightness over the course of the survey, we draw each simulated exposure from a stochastic distribution of these parameters. We then consider two different strategies to use the simulated exposures. First, we build one single-epoch image for each field (hereafter labeled as LSST-best) by keeping only the best seeing exposure. Second, we build another “worst-case” stacked image by degrading all individual exposures, ten per filter per year, to match the one with the worst seeing and co-add all exposures to a single image (hereafter labeled as LSST10). These two sets of images will allow us to investigate the trade-off between resolution and signal-to-noise for our automated lens search.

Figure 1 illustrates sample mock observations with a strong lensing signature from each telescope. The left-most column corresponds to a mock HST lensing system with a highly magnified source galaxy (top) and a less visible image of the source galaxy (bottom). For the HST-like dataset, many arcs are visually obvious due to the exquisite spatial resolution and quality of space based imaging.

The middle column corresponds to the same simulated systems for LSST10. The right-most column corresponds to the simulated systems of LSST-best. These images have resolution  $45 \times 45$  pixels<sup>2</sup> with 0.18 arcsec/pixel. LSST10 images visually exhibit the improved signal to noise ratio, recovering the arc feature, albeit at a much lower resolution than with the HST-like image or the LSST-best image. The top images of LSST10 and LSST-best show a visible lensed source galaxy image. The bottom images are washed out in the bottom row, where the magnification of the source galaxy is not as large. The ground based noise, PSF, and limited resolution of the LSST-best make visual giant arc identification difficult, except in systems with the most magnified source galaxies.

Our mock observations also include non-lens containing images. The procedure is similar to mock lensed images but we do not perform ray-tracing, so these images do not have lensed source galaxies.

Furthermore, we investigate the influence of light from the lens galaxies on the performance of our lens identification pipeline. We generate another set of each HST, LSST-best, and LSST10 images without the lens galaxy. We respectively label these nHST, nLSST-best, and nLSST10.

The final data set is then comprised of  $6 \times 10,000$  lens containing images, and  $6 \times 10,000$  non-lens containing images. We also keep a hold-out set of  $6 \times 1,000$  lens containing images, and  $6 \times 1,000$  non-lens containing images.

## 2.2. Strong Arc Lensing Identification Pipeline

To perform our analysis, we have used tools from *Scikit-learn* (Pedregosa et al. 2012). We outline the identification pipeline in Figure 2.2, which is a general description of supervised classification. Supervised classification is a class of machine learning where the class labels in the training set are known. In our case, the labels are “lens” and “non-lens” containing images.

The first step of our pipeline consists of a feature extraction stage, where our feature vector is a *histogram of oriented gradients* (HOG) (Dalal & Triggs 2005) that quantifies edges in the image. We describe the method and parameter search in Section 2.2.1. We then use *Logistic Regression* (LR), a machine learning algorithm described in Section 2.2.3, to train a classifier model on a subset of our images. LR requires a parameter search over the regression coefficient,  $C_{LogReg}$ ,

which we explore in Section 3.2. We briefly comment that our initial tests with a *Support Vector Machine* (SVM) using radial basis functions as an alternative machine learning algorithm yielded negligible performance improvement, and significantly increased computation cost. This indicated that the features of lens and non-lens images are relatively well separated by hyperplanes in feature space. For these reasons, we do not include SVM in our final analysis and comparisons and continue all discussions with a linear classifier.

Both the feature extractor, HOG, and the linear classifier, LR, contain parameters, which must be tested and optimized for peak model performance. We use *GridSearchCV* from *Scikit-learn* to select cross-validated parameters, and discuss this step of our methodology in Section 2.2.2.

The second step of our analysis is to test our trained model on an independent subset of the images to assess the model performance. Here, we evaluate the model on each test image, predicting a likelihood (“score”) between 0 and 1 that image contains a lensing system. This “holdout set” is not used in any of our parameter searches to keep our test metric independent of tuning.

### 2.2.1. Feature Extractor: Histogram Oriented Gradients

Originally created for human detection in computer vision, histogram of oriented gradients (HOG) is a feature extraction method that computes centered horizontal and vertical gradients. HOG is relatively robust to noise in the image, and is a fairly fast transform that describes edges. Details can be found in Dalal & Triggs (2005), but we describe the procedure here. The end result of HOG is a one dimensional histogram computed as follows.

HOG first divides the image into blocks of 50% overlap. Each block contains  $m \times m$  cells-per-block that each contain  $n_{pix} \times n_{pix}$  pixels-per-cell. The computed gradient orientation is quantized into  $N_{orient}$  bins.

The orientation gradient of all pixels within each cell are binned into the quantized orientations, providing a net gradient description within that cell. As an example, for  $N_{orient} = 3$ , our bins are centered at  $\theta = 0, 2\pi/3, 4\pi/3$  in radians. If a cell only has a gradient in the  $\theta = \pi/2$  direction, it will contribute 75% of its magnitude to the  $\theta = 2\pi/3$  bin, and 25% of its magnitude to the  $\theta = 0$  bin. The bins in all cells are then concatenated to make a larger feature vector that is  $N_{orient} \times N_{cells}$ .

The last step is a normalization procedure to control for illumination effects. Here, the sub-histograms of each cell within the same block are normalized with respect to one another before the transformation returns the final feature vector. The division of the image and the quantization of orientations are thus controlled by 3 parameters in HOG:  $N_{orient}$ , cells-per-block, and pixels-per-cell. We discuss how we select parameters using cross-validation in Section 2.2.2

### 2.2.2. Optimized Pipeline Parameters with a Grid Search

We run a grid search across parameters that should reasonably sample the arc edges in either the HST- or LSST-like mock observations, and illustrate the results in Table 1. Recall, the HST-like images are  $300 \times 300$  pixels per image, while the LSST-like mock observations are  $45 \times 45$  pixels per image.

We first estimate the size of a cell that will contain a coherent arc feature. To first order approximation, subdivisions of cells that are 1/100th the area of the entire image should contain coherent arc edges that span an elongated shape within

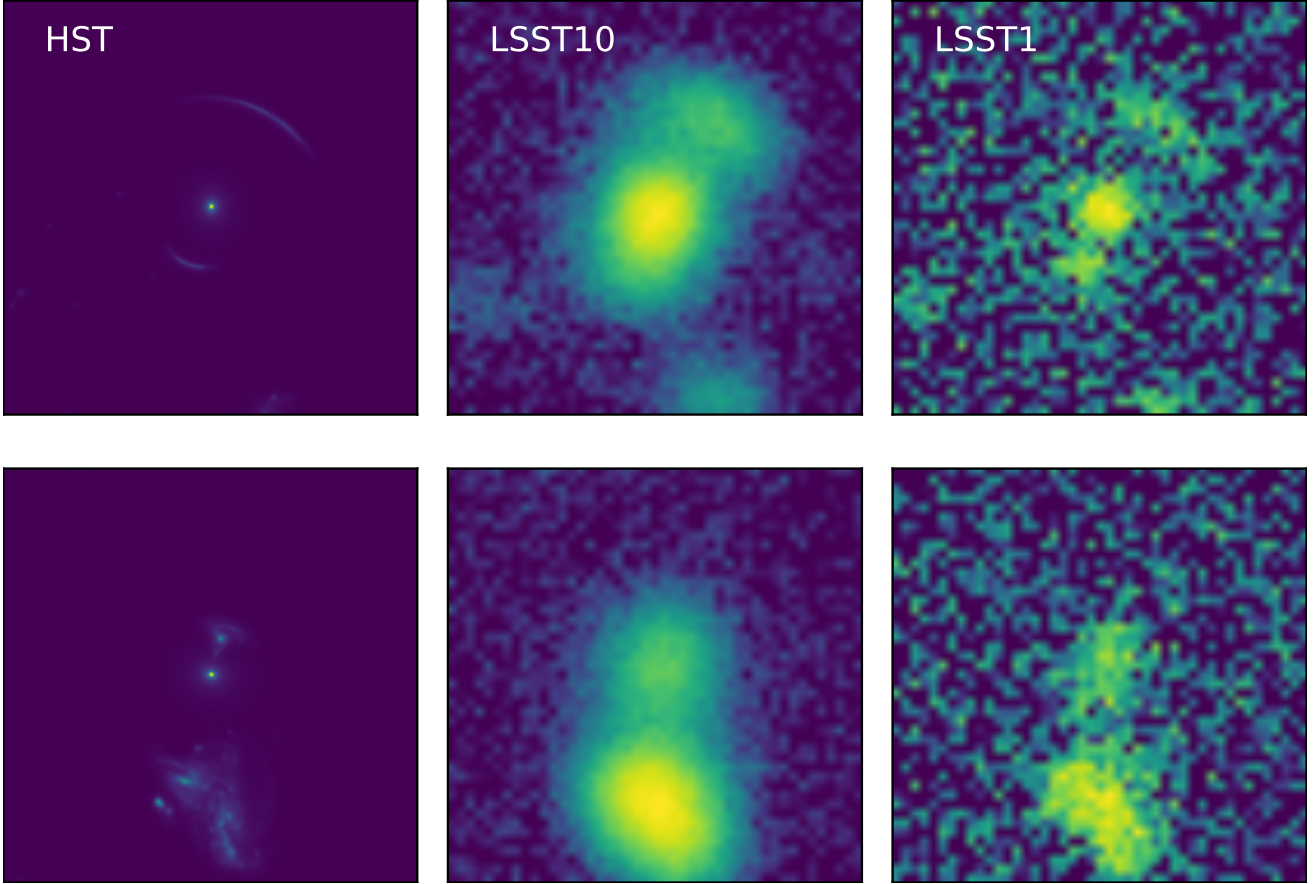


FIG. 1.— Left to right show example mock HST, LSST 10 year, and LSST-best images. The top row corresponds to a lensing system with a very visible arc signature, and the bottom row to a lensing system that is less obvious. Example mock HST images have  $n_{\text{pix}} \times n_{\text{pix}} = 300 \times 300$ . Example mock LSST images have  $n_{\text{pix}} \times n_{\text{pix}} = 45 \times 45$ . The resolution and noise of a ground based telescope is noticeably worse. Visual identification of giant arcs in the LSST images in the bottom row is very difficult.

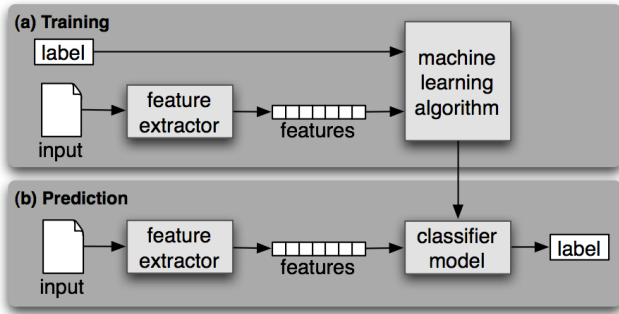


FIG. 2.— Cartoon of pipeline <sup>a</sup> for supervised classification. In our case, the labels are either lens or non-lens containing images. Our input is the set of mock HST- or LSST-like observations. The feature extractor is the histogram of oriented gradients, producing an N-dimensional feature vector that quantify edges in our images. Our machine learning algorithm is logistic regression.

<sup>a</sup>from <http://www.nltk.org/book/ch06.html>

arc-containing cells. Therefore, we sample the pixels per cell parameter from (8, 8) to (32, 32) for the HST-like images, and (3, 3) to (5, 5) for the LSST-like images in our grid search.

Next, the cells per block parameter determines the normalization of each cell with respect to the neighboring cell. In general, this will down-weight arc-like edges in cells that neighbor very bright cells, such as cells that cover the central lens galaxy. We therefore vary the cells per block param-

eter between (2, 2) and (4, 4) for the LSST-like images and between (1, 1) and (4, 4) for the HST-like images.

The number of orientations will determine the sampling of rounded edges. For example, if we only have two orientations, an arc-like feature in a cell directly north of the lensing galaxy will appear in our HOG visualization as a strong horizontal line (e.g. see top left in Figure 3), and an arc-like feature north-east of the lensing galaxy will appear as an L-shape. However, contributions from a cluster or line-of-sight galaxy in the same cell will tend to contribute edges in all orientations of the histogram (e.g. bottom right in Figure 3).

Finally, the resolution of the overall image will also limit the additional information that an increase in  $N_{\text{orient}}$  will provide. From the grid search, the best case number of orientations for each dataset is  $N_{\text{orient,HST}} = 9$ ,  $N_{\text{orient,LSST10}} = 3$ , and  $N_{\text{orient,LSST-best}} = 5$ .

The image resolution affects the length of the HOG feature vector, which has a monotonically increasing relationship with the time required to train the model. Additionally, for fixed memory restrictions, there is a tradeoff between the length of the feature vector and the size of the training set. We will discuss how the training set size affects the train time for each data set in Section 3.3.1.

### 2.2.3. Machine Learning Algorithm: Logistic Regression

The problem of detecting gravitational lenses in images falls under the general category of *classification* in machine



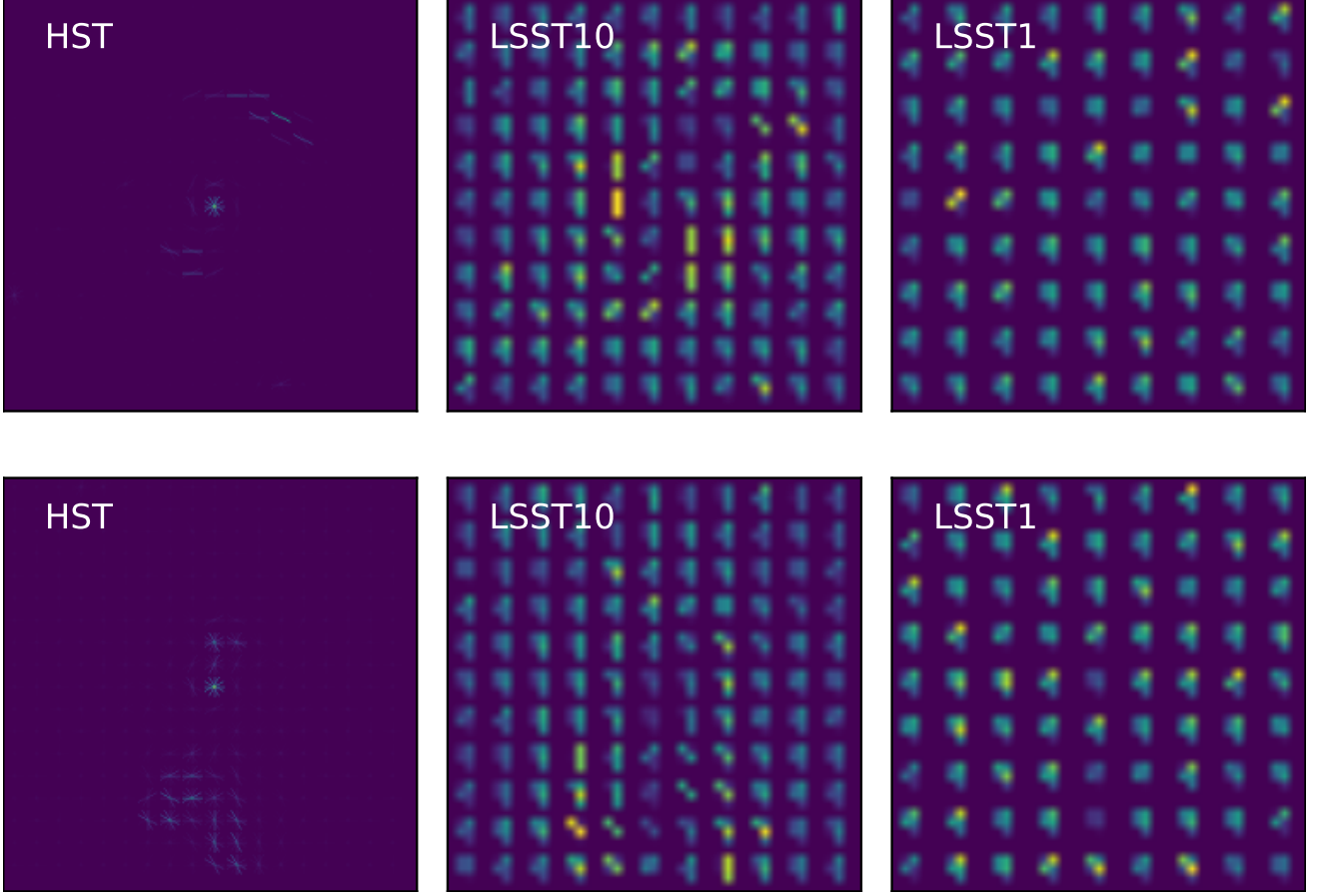


FIG. 3.— Left to right show example image transforms of mock images from Figure 1 with a visualized histogram of oriented gradients. The image transform picks up edge features, with arc features showing up as edges across radial orientations. Each of the oriented gradients within a cell is color-coded by magnitude, and represented as a line in the direction perpendicular to that gradient. The actual extracted features fill a one-dimensional feature vector comprised of the magnitudes of each of the oriented edges within the visualized cells.

learning. In general, the task is to find a function that assigns data points  $x$  to one of two or more classes, denoted by the class label  $y$ . This is equivalent to specifying a decision boundary, or decision boundaries between the classes in the space of the data points. (Compare this to *regression* in which the task is to find a function  $y = f(x)$ , where  $y$  is a continuous, rather than discrete, variable.) In our case, we have two classes: lens and non-lens containing images, and the data points  $x$  are the HOG feature vectors extracted from the images. In this paper we use the *Logistic Regression* (LR) algorithm, for which the decision boundary is a hyperplane. (The equivalent in the regression setting would be linear regression.) In LR we determine the optimal hyperplane by minimizing the objective function

$$L(A, b) = \sum_i \log [1 + \exp(-y_i(A \cdot x_i + b))] \quad (5)$$

where  $x_i$  is a data point (HOG feature vector),  $y_i$  is the known label for that data point (1 for a lens containing image,  $-1$  for a non-lens containing), and  $A$  and  $b$  are the parameters of the hyperplane. Eq. 5 is to be minimized with respect to  $A$  and  $b$ . Other more complicated machine learning algorithms exist which do not necessarily produce a linear decision boundary, such as *Support Vector Machines* (SVM), *Random Forests*, and *Neural Networks* (Hastie et al. 2009).

The HOG feature vectors in this paper can be very high-

dimensional. When dealing with high-dimensional data, where the number of dimensions becomes comparable to the number of data points, overfitting can become an issue. (An example of an extreme case of overfitting is fitting a degree  $n$  polynomial to  $n$  points. The polynomial would simply wiggle so that it goes through every point, and would have no predictive power if you tried to interpolate or extrapolate.) In machine learning, overfitting is avoided using different *regularization* techniques. A common choice for logistic regression is to add a penalty term to Equation 5

$$L_{\text{Reg}}(A, b, C_{\text{LogReg}}) = L(A, b) + \frac{1}{2C_{\text{LogReg}}} \|A\| \quad (6)$$

where the norm  $\|A\|$  is typically taken as either the sum of squares of the coefficients ( $L_2$  norm) or the sum of absolute values of the coefficients ( $L_1$  norm). In this paper we use the former.

The amount of regularization is controlled by the parameter  $C_{\text{LogReg}}$ : larger values of  $C_{\text{LogReg}}$  correspond to increasing model complexity. If  $C_{\text{LogReg}}$  is too large then the model will overfit, and if it is too small the model will underfit. To determine whether a model is overfit or underfit, the model is trained, (i.e. Eq. 6 is minimized), on a subset of the data called the *training set* and its performance (goodness of fit) is evaluated both on the training set and a separate *test set* that was not used in constructing the model. Figure 6 shows the

TABLE 1  
GRID SEARCH OF PIPELINE PARAMETERS

$N_{\text{orient}}$	Pixels/Cell	Cells/Block	$C_{\text{reg}}$	Score
(a) HST-like data				
9	(16, 16)	(4, 4)	10.	$0.90141 \pm 0.00658$
9	(24, 24)	(3, 3)	10.	$0.89266 \pm 0.01033$
9	(16, 16)	(1, 1)	10.	$0.87484 \pm 0.00671$
7	(16, 16)	(4, 4)	10.	$0.90031 \pm 0.00956$
5	(16, 16)	(4, 4)	10.	$0.89453 \pm 0.00840$
9	(32, 32)	(4, 4)	10.	$0.85828 \pm 0.01004$
9	(32, 32)	(4, 4)	50.	$0.87344 \pm 0.01267$
(b) LSST-like data				
3	(5, 5)	(4, 4)	100.	$0.64328 \pm 0.00769$
4	(5, 5)	(4, 4)	100.	$0.62469 \pm 0.00485$
3	(5, 5)	(2, 2)	100.	$0.63281 \pm 0.00869$
5	(5, 5)	(4, 4)	100.	$0.62734 \pm 0.00800$
3	(3, 3)	(4, 4)	100.	$0.59109 \pm 0.00313$
3	(3, 3)	(2, 2)	100.	$0.57297 \pm 0.00508$
4	(4, 4)	(3, 3)	100.	$0.63125 \pm 0.01227$

NOTE. — Panel (a) shows a subsample of the results of a grid search for HST across a range of HOG parameters and regression coefficient. Panel (b) shows a subsample of the results of a grid search for LSST10. Each use a data set of size  $2 \times 8000$  for cross validation to get the average scores and standard deviation. We explore different HOG parameters in each dataset due to resolution and image size differences.

performance of a model with selected HOG parameters as a function of the regularization parameter,  $C_{\text{LogReg}}$ , for both the training and the test set. (The performance can be measured by the accuracy, i.e. percent of images correctly classified, or by some other metric, such as the area under the Receiver Operating Characteristic curve described in Section 3.1.)

When  $C_{\text{LogReg}}$  is small, the performance of the model improves with increasing  $C_{\text{LogReg}}$  on both the training and test set, meaning that  $C_{\text{LogReg}}$  is still in the underfitting regime. Eventually the performance on the test set reaches a maximum and starts to decrease, even while the performance on the training set continues to increase. This means that the model is no longer generalizing well and is starting to overfit. The optimal  $C_{\text{LogReg}}$  occurs when the performance of the test set is at its maximum; this is the value of  $C_{\text{LogReg}}$  that should be used in the final model.

In practice, there is something of a trade-off between accuracy and computational resources because a larger value of  $C_{\text{LogReg}}$  will also increase the training time, since a larger  $C_{\text{LogReg}}$  corresponds to a less constrained parameter space being searched. We discuss the performance and training time dependence on  $C_{\text{LogReg}}$  in Section 3.2.

### 3. RESULTS

#### 3.1. Receiver Operating Characteristic

In this section, we discuss the Receiver Operating Characteristic (ROC) curve (see Figure 4), which shows the true positive rate (tpr) as a function of false positive rate (fpr) for a given model and test set. The true positive rate is defined as the number of true positives divided by the total number of positives (completeness of the positive identifications). The false positive rate is defined as the number of false positives divided by the total number of negatives (impurity of the identifications). The ROC curve illustrates the performance of our trained model as we vary the discrimination threshold.

The classifier model assigns a score to each test image, which is a probability that the image is a strong lensing system. To construct the ROC curve, we rank the test images by probability, and calculate the tpr and fpr for decreasing discrimination threshold.

For very high discrimination threshold, we have maximal purity, but a low completeness (bottom left region of the ROC). For a very low discrimination threshold, we have maximal completeness, but minimal purity (top right region of the ROC). The ideal model would have an ROC curve with data points that go from  $(x,y) = (0,0)$  to  $(x,y) = (0,1)$  to  $(x,y) = (1,1)$ .

In the context of strong lensing systems, we wish to maximize the true positive rate so we have a representative count of the fraction of strong lensing systems in an observed volume of the universe. We also want to minimize the false positive rate. Positively identified strong lensing systems will require expensive spectroscopic follow-up for validation. The steepness of the ROC curve indicates how well the model will optimize the two, and we can characterize the steepness by the area under the curve (AUC). The ideal model would have an AUC of 1. We show the ROC curves of our best performing models in each dataset.

For comparison, we also show a Precision-Recall curve (see Figure 5), where precision is the number of true positives identified as positive divided by the total number of positively identified images (purity) and recall is the number of true positives identified as positive divided by the total number of true positives (completeness). Each point in the figure is calculated with a varying threshold for identification. Since our sample has a class balance of 50-50 between positives (lens) and negatives (non-lens), the most lenient threshold for identification would yield a precision of 0.5 at a recall of 1.0. It is important to note that this figure changes as the class balance changes; if we had 90% non-lenses and 10% lenses, the most lenient threshold would yield a precision of 1/9 at a recall of 1. The ROC curve is a more typical metric for supervised classification.

Figure 4 shows the ROC curves for models trained using the entire 10,000 training sample, with best-case HOG and regularization parameters. The models have been evaluated on a hold-out set of 1,000 images that were not used in the parameter search. We show the mock HST, LSST-best, and LSST10 results respectively in red, blue, and green. Solid lines correspond to a model trained and tested on images with the lensing galaxy. Dashed lines correspond to a model trained and tested on images where the lensing galaxy is excluded from the mock observation, simulating an ideal modeling and subtraction of the lensing galaxy, which has been one proposed method to improve the identification of strong lensing systems. The corresponding AUC is listed in the legend.

The model performance for the mock HST data is  $\text{AUC}=0.975$  for images with the lens galaxy, and  $\text{AUC}=0.98$  for images without the lens galaxy (red solid and dashed). On

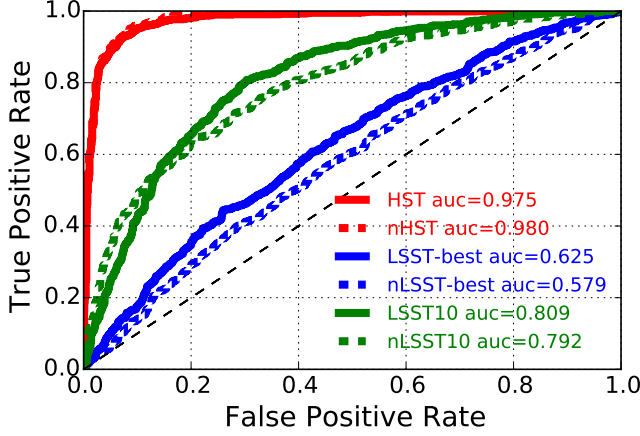


FIG. 4.— Red, blue, green: ROC curves for models trained on our whole 10,000 training set and tested on our holdout set of 1,000. These respectively correspond to the HST, LSST 1 year, and LSST 10 year data. The solid lines are for data that include the lensing central galaxy, and the dashed lines for the data where there is no lensing galaxy, mimicking an ideal removal of the lens. Model performance can be summarized by the area under the curve (AUC), labeled in the legend.  $AUC = 1$  is a perfect model, and  $AUC = 0.5$  is a useless model.

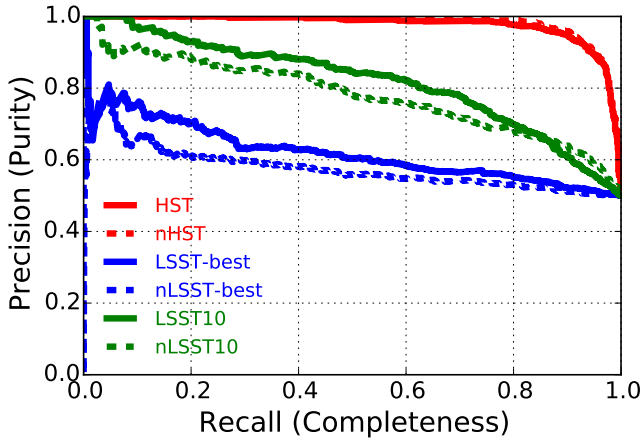


FIG. 5.— Red, blue, green: Precision-recall (PR) curves for models trained on our whole 10,000 training set and tested on our holdout set of 1,000. These respectively correspond to the same models and data shown in Figure 4. An ideal model would reach both a precision (purity) and recall (completeness) that equal 1. Note, that this performance describes a data set with a 50-50 split between lens and non-lens containing images.

the other hand, the model for our LSST-like dataset for one year has an  $AUC=0.625$  with the lens galaxy and  $AUC=0.579$  without the lens galaxy (blue solid and dashed), and the model for our LSST-like dataset for 10 years has an  $AUC=0.809$  with the lens galaxy and  $AUC=0.792$  without the lens galaxy (green solid and dashed). Removal of the lens galaxy does not systematically perform better, and is actually dependent on the size of the training set. We discuss relative model performance and complexity for images with and without the lens galaxy in Section 3.3.1.

### 3.2. Effects of Regularization on Model Performance

As described in Section 2.2.3, LR trains a model with complexity determined by the regularization parameter coefficient,  $C_{\text{LogReg}}$ . Larger values of  $C_{\text{LogReg}}$  are less regularized and allow for increased model complexity. The highest

values of  $C_{\text{LogReg}}$  will better describe features in the training set. However, an overly complex model will overfit the training set at the expense of its performance on any independent test set. The regularization parameter ultimately defines the model performance, and we must perform a parameter search to identify the optimal value for  $C_{\text{LogReg}}$ .

Figure 6 shows the model performance as a function of regularization parameter for each data set HST, LSST-best, and LSST10. The solid and dotted blue lines respectively correspond to the model performance on the test and training set, with the AUC as a metric for performance. In red, we show the train time as a function of  $C_{\text{LogReg}}$ .

As expected, the training set AUC increases and asymptotes with  $C_{\text{LogReg}}$ . With increasing model complexity, the model better fits the training data set. This is analogous to fitting a seventh order polynomial to seven data points, where the fitting function will go through every point but will not likely predict additional points. With increasing model complexity, we are able to better capture features that are generally characteristic of strong lensing systems with arcs. However, past a certain  $C_{\text{LogReg}}$ , the model performance on the test set decreases or asymptotes, as it has overfit the training set. We use the scaling of AUC with  $C_{\text{LogReg}}$  when training 8,000 out of our full 10,000 training data set to determine the best value for  $C_{\text{LogReg}}$ . However, the optimal parameter is also dependent on the size of the training set (see Section 3.3.1), so this choice is not generalizable.

For fixed training set size, the log of the train time roughly scales linearly with the log of  $C_{\text{LogReg}}$ . Since lower values of  $C_{\text{LogReg}}$  correspond to a more regularized model, there is a smaller volume in hyperparameter space to search for the best fit coefficients. The solution, on average, will converge more quickly, for more regularized models. The scaling is not purely monotonic because the fitting still has some randomness associated with the path it takes to convergence.

### 3.3. Dataset Size Dependence

#### 3.3.1. Effects of Training Set Size on Performance and Train Time

In this section, we show the effects of training set size on model performance on the hold-out test set of 1,000 images. We also compare the improvement between images that include the lens galaxy and images with no lens galaxy.

Figure 7 shows how the AUC depends on the log of the size of the training set for both the LSST10 (LSST-best) data in the solid blue (red) line, and the nLSST10 (nLSST-best) data in dashed blue (red) line. The AUC for models trained on the LSST10 data improves almost linearly with the log of the training set size, increasing from  $AUC=0.705$  to  $AUC=0.788$  when the train size is increased from  $2 \times 500$  lens/non-lens images to  $2 \times 8000$  lens/non-lens images. However, for the nLSST10 data, where the lens galaxy has been removed from the images, the improvement is less dramatic. With the same increase in training size, the AUC for nLSST10 changes from just below 0.77 to just below 0.78.

In the LSST10 case, the trained model can incorporate the additional information of the edges from the lens galaxy, which is correlated with the lensing cross-section and likelihood of the image being a lensing system. The nLSST10 images do not contain this information, but provide cleaner signals of the lensed image for lensed images that occur close to the lens galaxy. The cleaner signal in nLSST10 allows for better model performance for smaller training data set sizes ( $N_{\text{train}} \sim 5,000$ ). However, models trained on LSST10 im-



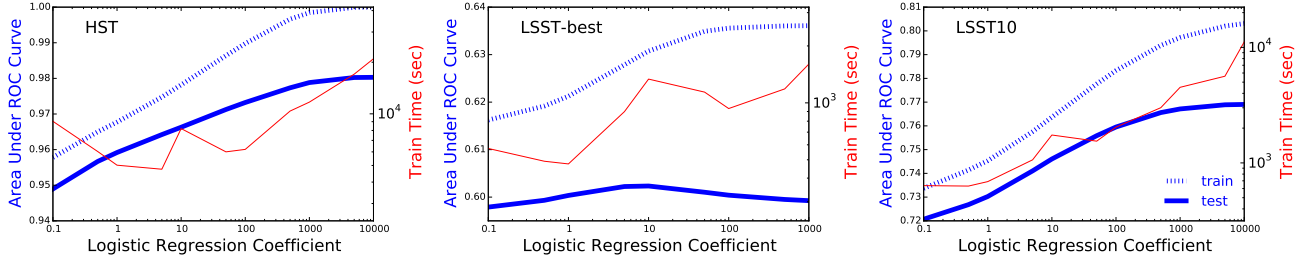


FIG. 6.— AUC of the model with varying LR regularization coefficient parameter,  $C_{\text{LogReg}}$ , used when training the model classifier. We use a subset of the 10,000 training images to search over the LR  $C_{\text{LogReg}}$  parameter, training on 8,000 and testing on 1,000. Each panel corresponds to a different mock observation. From left to right: HST, LSST for one year, and LSST for 10 years. The solid blue lines correspond to the AUC of the test set, and the dotted blue lines to the AUC of the training set. To avoid overfitting, we choose the smallest parameter for which the AUC of the test set is maximal: 5000, 10, and 5000, respectively. In thin red solid lines, we show the train time of the model, which roughly increases in a log-log scaling with logistic regression coefficient parameter.

prove more rapidly with train size, since the additional information from the lens galaxy better describes the lens containing images. For  $N_{\text{train}} \gtrsim 5,000$ , the models trained on LSST10 outperform those trained on nLSST10.

In red, we see an analogous improvement for LSST-best and nLSST-best. Here, the crossover happens at train size of  $2 \times 1000$ , where an increase in larger training data set size does not help improve the AUC for nLSST-best.

Since the models that contain information from the lens galaxy edges in LSST10 are more complex, the models require a larger training set size for a better fit. While model performance for LSST10 appears to steadily increase, this comes at the cost of increased train time, which is two-fold. The train time will increase due to both an increase in data to fit, and also an increase in optimal  $C_{\text{LogReg}}$  where the volume of hyperparameter space for allowed solutions is larger (see red lines in Figure 6).

We illustrate the dependence of train time on both the size of the training set and model complexity in Figures 8 and 9. In red, Figure 8 shows that the optimal values for  $C_{\text{LogReg}}$ . For LSST10,  $C_{\text{LogReg}}$  roughly scales logarithmically with the log of the train size, with an exception of the data point corresponding to train size of 2000. Generally, a larger training set allows for an increase in model complexity without reducing its ability to generalize. This is also true for the optimal  $C_{\text{LogReg}}$  dependence on the number of training images in the nLSST10 data. But, the required complexity is systematically less than for the LSST10 images.

In blue, Figure 8 also shows the train time for LSST10 and nLSST10 as a function of the size of the training set for the optimal regularization parameter for that subset of the training data. Each model uses features extracted with the same HOG parameterization from the grid search and the optimal regularization parameter for that subset of the training data. The train time of a given model generally increases for increasing regularization parameter. For the subset of train size  $N_{\text{train}} = 2000$  in LSST10, the optimal regularization parameter happened to be  $C_{\text{LogReg}} = 100$ , whereas it was  $C_{\text{LogReg}} = 500$  for the subset of train size 1000, and  $C_{\text{LogReg}} = 1000$  for the subset of train size 4000. This makes the train time increase at train size  $N_{\text{train}} = 2000$  for LSST10 less dramatic than the average log-log slope of approximately 2.

Since nLSST10 does not contain the lens galaxies, fewer of the extracted features describe the lens system, requiring decreased model complexity. The increase in train time for nLSST10 as a function of train size is mostly due to only having more data to fit in the regression, leading to a steady and slow increase of train time with number of training images

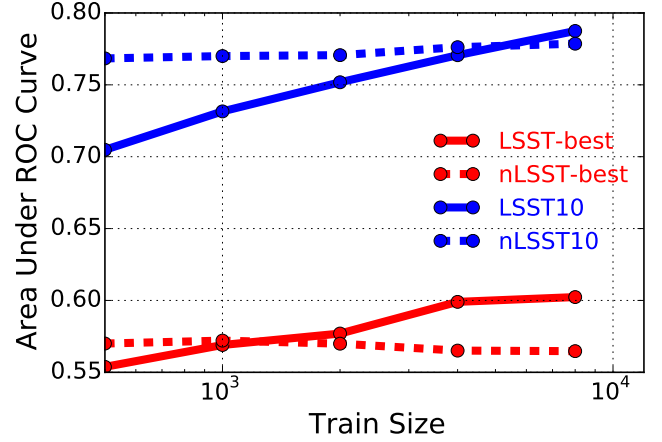


FIG. 7.— Solid (dotted) blue line: AUC for models with varying size of the training set for LSST10 (nLSST10). Solid (dotted) red line: Same for LSST-best. The improvement of AUC scales roughly linearly with the log of the training set size. However, LSST10 and LSST-best have a steeper improvement with data training size. The performance of nLSST10 and nLSST-best models trained on smaller training data sets is better than the respective LSST10 and LSST-best models trained on the same size data, but the LSST10 and LSST-best models outperform with larger size training data set.

with log-log slope of approximately 1.

For LSST-best, Figure 9 shows the same relationships between train time and train size in blue, and the optimal regularization parameter (or model complexity) in red. Contrary to what we found when comparing LSST10 to nLSST10, nLSST-best requires more model complexity than LSST-best. Recall that LSST-best and nLSST-best correspond to single epoch simulated exposures with the best seeing; these images exhibit better resolution but worse signal to noise. The switch in required model complexity corresponds to a trade-off between features from the lens galaxy providing additional information or swamping the signal from a strong lensing arc.

### 3.3.2. Effects of Rotation on AUC

To augment our training set, we rotated each image in the set by multiples of  $90^\circ$ . Since our feature extraction method of HOG is not rotationally invariant, augmenting our data by a factor of four naturally optimizes the use of available training data. This has an equivalent improvement to the study illustrated in Figure 7.

We also tested the effects of evaluating our model on all four rotations of the test set, and using the average score of each test image to calculate the AUC. In Figure 10, we show the AUC for different orientations of the datasets. The x-axis

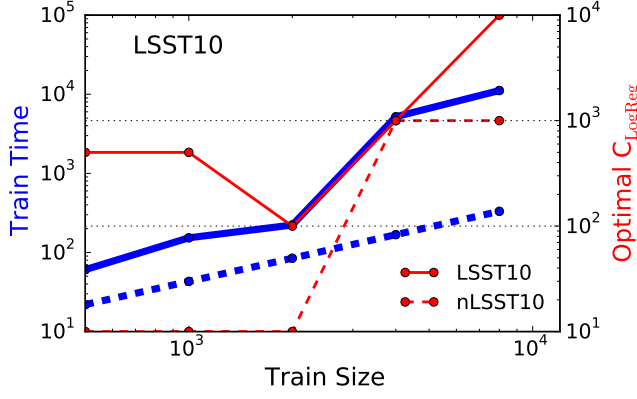


FIG. 8.— Solid (dotted) blue line: Train time for models with varying size of the training set for LSST10 (nLSST10). Train time roughly scales logarithmically with the train size, but the train time is also affected by model complexity. Solid (dotted) red line: Best regularization parameter as a function of train size. Note, LSST10 requires more model complexity to exceed the performance of nLSST10 (see blue solid and dotted lines in Figure 7), and therefore requires more training time for continual increase in performance.

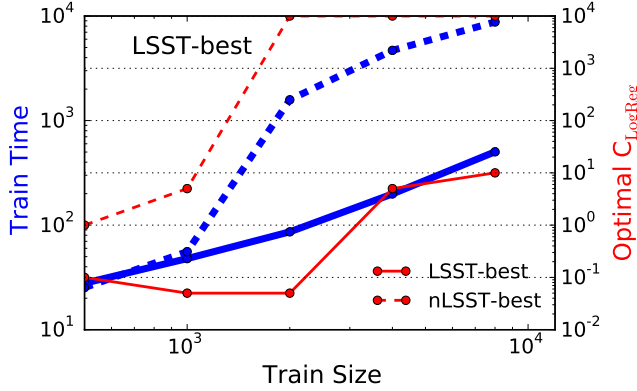


FIG. 9.— Same as Figure 8, but for LSST-best. Here, nLSST-best requires more model complexity than LSST-best. Note, that LSST-best and nLSST-best have better seeing but reduced overall signal to noise than LSST10 and nLSST10. This corresponds to better resolution, and therefore sharper edge features that prominently correspond to arcs.

corresponds to each of our three datasets, with and without the lens galaxy. The y-axis shows the AUC. The filled blue circles correspond to the four AUCs calculated when the model evaluated images at each of the four rotations. The filled red stars correspond to the AUC calculated from the average of all four test scores, which are systematically higher than any one rotation. The average score across all rotations for each image is likely to be less noisy for the whole test sample, giving an improved AUC.

Figure 10 also summarizes the best-case results of our models trained on our entire 10,000 training sets, and tested on our holdout 1,000 test set. Recall, however, that we expect model performance on images containing the lens galaxy to improve further with larger training sets (see Figure 7).

### 3.4. Image Classification Performance

#### 3.4.1. Populating the ROC Curve

In this section, we discuss the different image types that our model is most and least able to successfully classify. We have six paradigms of model performance on the mock images based on the score an image receives when evaluated by

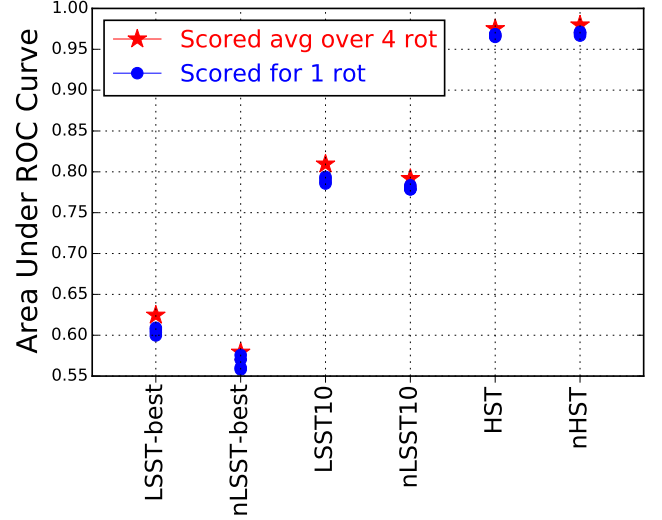


FIG. 10.— Our summary figure: The AUCs of models trained on the full 10,000 and tested on the holdout 1,000. Blue circles: AUC calculated from scores of images at a given rotation (e.g. 0, 90, 180, and 270 degrees). Red stars: AUC calculated from the average score of all rotations of each image. The average score produces an improved AUC in all data sets. We expect the AUC to further improve with increased train size.

the trained model, and its true label. From highest scoring to lowest scoring: True Positives (tp), False Positives (fp), Borderline Positives (bp), Borderline Negatives (bn), False Negatives (fn), and True Negatives (tn).

Figure 11 illustrates four images from each paradigm for the hold-out test set of LSST10. The trained model used the entire 10,000 image training set. The left two columns show lens systems, and the right two columns show non-lens systems. The annotation in the top left corner of each images shows the score.

In general, the true positives (the highest scoring lens systems) have lensed images with large magnification. The true negatives (the lowest scoring non-lens systems) have small lens galaxies with galaxies along the line of sight that are rounded. The successful classification of these two paradigms are least sensitive to the threshold. On the other hand the failed classification of the false positives (the highest scoring non-lens systems) and the false negatives (the lowest scoring lens systems) are also least sensitive to the threshold. False negatives are typically lens systems with lensed images of smaller magnification and minor distortions that mimic along the line of sight galaxies that the model has learned to ignore. False positives are often non-lens systems with elongated, elliptical, or “fuzzy”, galaxies along the line of sight whose signal blends with the lens galaxy contributing to the false positive rate even for conservative thresholds. Visually, these false positives are virtually indistinguishable from true arcs, and would require spectroscopic follow-up.

The middle two rows of Figure 11 illustrate the borderline positives and borderline negatives. The successful classification of the borderline positives and negatives are most sensitive to the threshold, and would be the first candidates for alternative classification methods, such as visual follow-up. Thresholds set around these scores yield a true positive rate and false positive rate of  $tpr \approx 0.8$  and  $fpr \approx 0.25$ , respectively.

#### 3.4.2. Dependence on Lens-Model Parameters

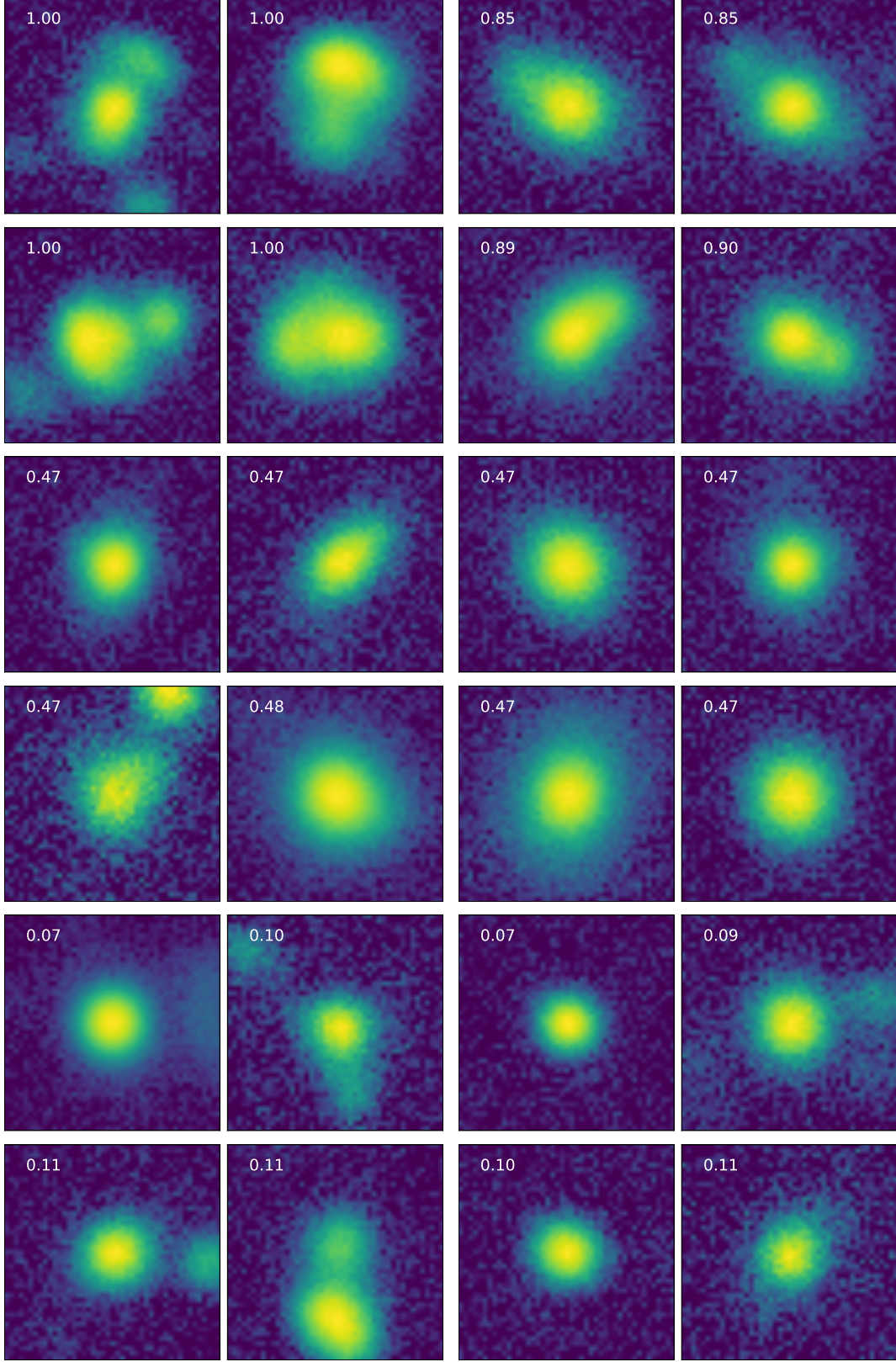


FIG. 11.— LSST 10 year mock images. Left two columns: Lens containing images, annotated with the image score assigned by our trained classifier. Right two columns: Non-lens containing images, annotated with the image score. The top two rows show characteristic images that will be accepted with a high threshold for classification, contributing to the bottom left of the ROC curve in Figure 4. The middle two rows show characteristic images that will be accepted with a moderate threshold, contributing to the knee of the ROC curve, with true positive rates and false positive rates of  $tpr \approx 0.8$  and  $fpr \approx 0.25$ , respectively. The bottom two rows show characteristic images that will only be accepted with an extremely lenient threshold, contributing to the top right area of the ROC curve.



Here, we examine lens-model parameters that affect how well our pipeline can classify the system. The lens-model parameters we examined are the redshift, ellipticity, orientation angle, and velocity dispersion of the lensing galaxy, and also the magnification of the lensed image compared with its original size. We found that the magnification of the lensed image is the most correlated lens-model parameter with our trained model performance, with a secondary and related correlation with lens galaxy velocity dispersion that is encapsulated in the Einstein radius. The more strongly lensed an image is, the larger its magnification, and the easier it is for our trained model to classify.

In Figure 12, we show the classification score as a function of Einstein radius and image magnification of the source galaxy in each of our samples. In an HST- or LSST10-like observation, lensing systems with images that have magnification  $\gtrsim 7$  will likely be classified as positive with threshold scores above  $\gtrsim 0.5$ . These systems are also those that are most easily classified by eye. However, our trained model has varying performance for systems with lower magnifications.

Highly magnified systems typically have lenses that are at higher redshifts of and/or lens galaxies with velocity dispersions larger than  $\sigma_v \gtrsim 230$  km/s. Lensing galaxies with smaller velocity dispersions are less massive and therefore have a smaller efficiency of lensing cross section. The smaller efficiency means that a background galaxy is less likely to be strongly lensed with high magnification. Also, due to hierarchical structure formation, galaxies are less massive at higher redshifts, so the trends of model performance with these three parameters are somewhat degenerate with one another.

The relationship between our model performance and lens parameters indicates that the magnification of lensed images is the most relevant, and the distribution of image magnification in a data set will impact trained model performance. We did not find strong correlations in model performance with other lens parameters. It is also useful to keep in mind that lensed galaxy images with magnification  $\lesssim 5$  are often visually indistinguishable from edge-on disk galaxies along the line of sight, which can lead to false positives. Since the latter can lead to false positives, the model has learned to down-weight the related features. Note that the model would be more sensitive to systems with lower magnification without galaxies along the line of sight.

In a forthcoming paper, we will discuss how class imbalance, or differences between the lens-model parameter distributions in the training and test sets affect model performance and will explore a method to correct for this.

#### 4. SUMMARY AND DISCUSSIONS

We have presented a supervised classification pipeline to automatically identify galaxy-galaxy strong lensing systems using a histogram of oriented gradients (HOG) as a feature extractor, and logistic regression (LR) as a machine learning algorithm. Our pipeline can easily be extended to identify other strong lensing features, such as multiply imaged quasars, and to test alternative features and/or machine learning algorithms.

We have also made use of a new sophisticated set of mock observations, which will be made publicly available. The lensing systems have lens galaxies generated with a realistic redshift distribution and along the line of sight galaxies drawn from Hubble Ultra-Deep field observations. We have explored results from mock HST, one year LSST, and ten year LSST observations.

We summarize key points below:

- We have designed our pipeline to easily select and add image pre-processing and feature extraction methods, and to select a machine learning algorithm for classification. Additionally, the user can easily perform parameter searches to train a model with the best parameters for a given problem.
- We have tested and run parameter tests for a histogram of oriented gradients as an efficient and effective feature extractor for galaxy-galaxy strong lensing systems in both a space based (HST-like) and ground based (LSST-like) observation. We have also tested and run parameter tests for logistic regression as a scalable, cheap, and effective machine learning algorithm.
- We find AUC values of ROC curves of optimized classifier models to yield  $AUC = 0.975$  for the HST-like data,  $AUC = 0.809$  for the stacked LSST-like data. Model performance exhibits continual increase with the training size.
- While removal of the lens galaxy improves model performance for smaller size training samples, features from the lens galaxy improve model performance for larger training data sets.
- Images that were easiest for our model to classify typically were lens systems that had high lensed image magnification and a lens galaxy with large velocity dispersion or non-lens systems with lens galaxies with smaller velocity dispersion and non-elongated along the line of sight galaxies.

We emphasize that simple linear classifiers, such as logistic regression, are scalable and relatively easy to parallelize with open source tools such as *Apache Spark*<sup>4</sup>. Our work indicates that HOG feature extraction plus a linear classifier captures much of the complexity in the arc finding problem. *These results indicate that effective image classification of galaxy-galaxy strong lensing systems can be achieved without large scale computing resources, or specialized hardware like GPUs.* The methods also scale to large datasets on a computing cluster, if needed.

One major caveat to our results is the fact that our mock data likely does not describe the full range of lens and non-lens images that will be observed. For example, the along the line of sight galaxies in our training and test images all come from the Hubble Ultra Deep field, sampling a smaller range of potential contaminants that are not associated with a lensed image. A limitation of the CANDELS sources is that the sources are observed with HST seeing, and would not resolve the clumpiness within a true HST arc. Also, the redshift of the source galaxies have been fixed to  $z_s = 2$ . We will address these shortcomings in subsequent work.

Finally, we must take the class balance, or relative number of lens to non-lens systems, into account when assessing a metric. The Purity-Recall (purity-completeness) metric is sensitive to the ratio of lens and non-lens systems in the data. Both of our training and test data sets have 50% lensed and 50% unlensed images, which is not expected in observations.

<sup>4</sup> <http://spark.apache.org/>

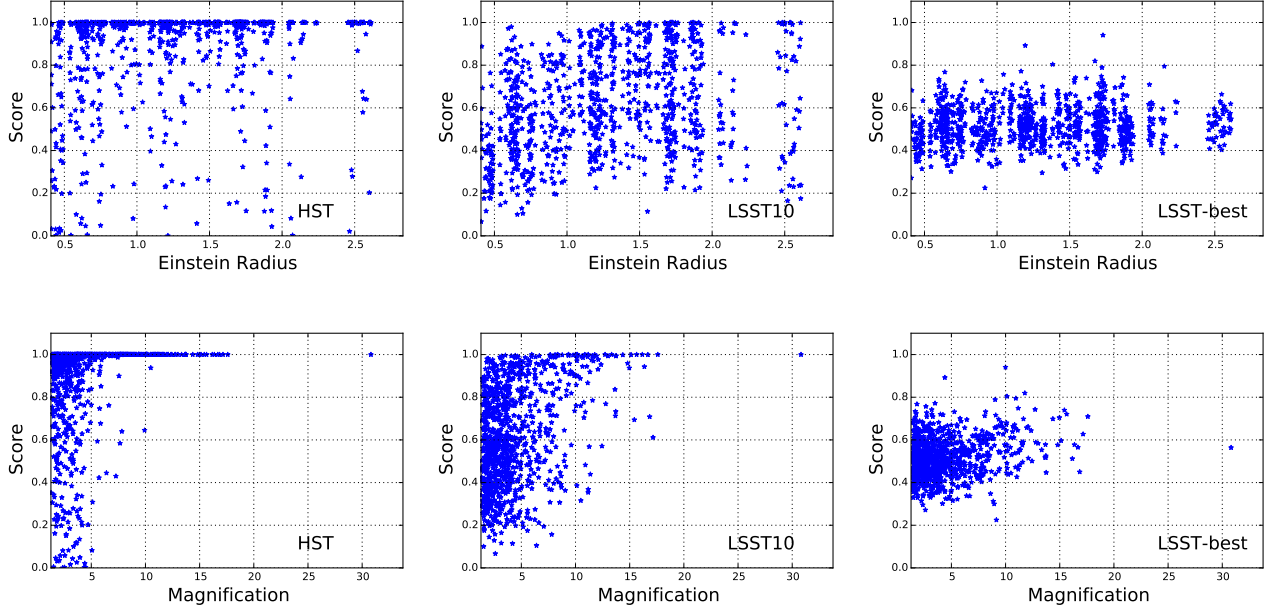


FIG. 12.— Left to right: Image classification score on HST, LSST10, LSST-best mock observed lensing images as a function of Einstein radius (top) and magnification (bottom). Magnification is the strongest indicator of how likely a lensing system will be successfully classified. The Einstein radius has a weaker correlation with how easily a lensing system might be classified, since it contains information on both the velocity dispersion of the lens and the redshifts of the lens and source galaxies. High velocity dispersion lens galaxies are likely to produce high magnification images of the source galaxies.

For images selected for a massive elliptical, the number of non-lenses will outnumber lenses by at least 1000 to 1. Therefore, a sample that is 50% pure requires a classifier with a false positive rate of 0.001. Looking at the solid green line in Figure 5, we can set a high classification threshold and obtain a sample with close to 100% purity and up to  $\sim 10\%$  recall of all of our lenses, before contamination from non-lenses leaks into the selection. While information from other bands will certainly improve the model performance, a maximally large and pure sample from our method would likely require further filtering, e.g. by citizen science.

The ROC curve metric is insensitive to the ratio, but is sensitive to the sampling. Given alternative lens and non-lens sample splittings, our true positive and false positive rates in the ROC curves would stay the same, making the ROC curve a more standard metric in the literature. On the other hand,

the ROC curves show a representative rate for lens and source distributions that are evenly sampled. We do *not* expect this sampling to be representative of what we might expect from an observational survey. We leave these additional challenges to future work.

CA acknowledges support from the Enrico Fermi Institute at the University of Chicago, and the University of Chicago Provost's Office. NL would like to thank the funding support from NSFC, grant no. 11503064, and Shanghai Natural Science Foundation, grant no. 15ZR1446700. This work was also supported in part by the Kavli Institute for Cosmological Physics at the University of Chicago through grant NSF PHY-1125897 and an endowment from the Kavli Foundation and its founder Fred Kavli.

## REFERENCES

- Agnello, A., Kelly, B. C., Treu, T., & Marshall, P. J. 2015, *MNRAS*, 448, 1446
- Alard, C. 2006, *ArXiv Astrophysics e-prints*, astro-ph/0606757
- Allam, S. S., Tucker, D. L., Lin, H., et al. 2007, *ApJ*, 662, L51
- Bezacourt, J., Pello, R., & Soucaill, G. 1998, *A&A*, 330, 399
- Bom, C. R., Makler, M., Albuquerque, M. P., & Brandt, C. H. 2016, *ArXiv e-prints*, arXiv:1607.04644
- Bonvin, V., Courbin, F., Suyu, S. H., et al. 2017, *MNRAS*, 465, 4914
- Braut, F., & Gavazzi, R. 2015, *A&A*, 577, A85
- Chae, K.-H. 2003, *MNRAS*, 346, 746
- Collett, T. E. 2015, *ApJ*, 811, 20
- Collett, T. E., & Auger, M. W. 2014, *MNRAS*, 443, 969
- Collett, T. E., Auger, M. W., Belokurov, V., Marshall, P. J., & Hall, A. C. 2012, *MNRAS*, 424, 2864
- Connolly, A. J., Peterson, J., Jernigan, J. G., et al. 2010, in *Proc. SPIE*, Vol. 7738, Modeling, Systems Engineering, and Project Management for Astronomy IV, 77381O, doi:10.1117/12.857819
- Dalal, N., & Triggs, B. 2005, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*
- Dieleman, S., Willett, K. W., & Dambre, J. 2015, *MNRAS*, 450, 1441
- Dye, S., Evans, N. W., Belokurov, V., Warren, S. J., & Hewett, P. 2008, *MNRAS*, 388, 384
- Estrada, J., Annis, J., Diehl, H. T., et al. 2007, *ApJ*, 660, 1176
- Galametz, A., Grazian, A., Fontana, A., et al. 2013, *ApJS*, 206, 10
- Gavazzi, R., Marshall, P. J., Treu, T., & Sonnenfeld, A. 2014, *ApJ*, 785, 144
- Gavazzi, R., Treu, T., Rhodes, J. D., et al. 2007, *ApJ*, 667, 176
- Gladders, M. D., Hoekstra, H., Yee, H. K. C., Hall, P. B., & Barrientos, L. F. 2003, *ApJ*, 593, 48
- Grogin, N. A., Kocevski, D. D., Faber, S. M., et al. 2011, *ApJS*, 197, 35
- Hastie, T. J., Tibshirani, R. J., & Friedman, J. H. 2009, *The elements of statistical learning : data mining, inference, and prediction*, Springer series in statistics (New York: Springer)
- Joseph, R., Courbin, F., Metcalf, R. B., et al. 2014, *A&A*, 566, A63
- Jullo, E., Natarajan, P., Kneib, J.-P., et al. 2010, *Science*, 329, 924
- Kneib, J.-P., & Natarajan, P. 2011, *A&A Rev.*, 19, 47
- Kochanek, C. S. 1996, *ApJ*, 473, 595
- Koekemoer, A. M., Faber, S. M., Ferguson, H. C., et al. 2011, *ApJS*, 197, 36



- Koopmans, L. V. E., Treu, T., Bolton, A. S., Burles, S., & Moustakas, L. A. 2006, *ApJ*, 649, 599
- Kubo, J. M., & Dell’Antonio, I. P. 2008, *MNRAS*, 385, 918
- Lanusse, F., Ma, Q., Li, N., et al. 2017, ArXiv e-prints, arXiv:1703.02642
- Lee, C.-H. 2017, ArXiv e-prints, arXiv:1702.07557
- Lenzen, F., Schindler, S., & Scherzer, O. 2004, *A&A*, 416, 391
- Li, N., & Chen, D.-M. 2009, *Research in Astronomy and Astrophysics*, 9, 1173
- Li, N., Gladders, M. D., Rangel, E. M., et al. 2016, *ApJ*, 828, 54
- Linder, E. V. 2004, *Phys. Rev. D*, 70, 043534
- Lintott, C. J., Schawinski, K., Slosar, A., et al. 2008, *MNRAS*, 389, 1179
- Lynds, R., & Petrosian, V. 1986, in *BAAS*, Vol. 18, *Bulletin of the American Astronomical Society*, 1014
- Marshall, P. J., Hogg, D. W., Moustakas, L. A., et al. 2009, *ApJ*, 694, 924
- Marshall, P. J., Verma, A., More, A., et al. 2016, *MNRAS*, 455, 1171
- Maturi, M., Mizera, S., & Seidel, G. 2014, *A&A*, 567, A111
- Miralda-Escude, J., & Lehar, J. 1992, *MNRAS*, 259, 31P
- More, A., Verma, A., Marshall, P. J., et al. 2016, *MNRAS*, 455, 1191
- Oguri, M., & Marshall, P. J. 2010, *MNRAS*, 405, 2579
- Paraficz, D., Courbin, F., Tramacere, A., et al. 2016, ArXiv e-prints, arXiv:1605.04309
- Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2012, ArXiv e-prints, arXiv:1201.0490
- Petrillo, C. E., Tortora, C., Chatterjee, S., et al. 2017, ArXiv e-prints, arXiv:1702.07675
- Seidel, G., & Bartelmann, M. 2007, *A&A*, 472, 341
- Sharon, K., Ofek, E. O., Smith, G. P., et al. 2005, *ApJ*, 629, L73
- Suyu, S. H., Treu, T., Hilbert, S., et al. 2014, *ApJ*, 788, L35
- Suyu, S. H., Bonvin, V., Courbin, F., et al. 2016, ArXiv e-prints, arXiv:1607.00017
- Walsh, D., Carswell, R. F., & Weymann, R. J. 1979, *Nature*, 279, 381
- Warren, S. J., & Dye, S. 2003, *ApJ*, 590, 673
- Xu, B., Postman, M., Meneghetti, M., et al. 2016, *ApJ*, 817, 85
- Zahid, H. J., Damjanov, I., Geller, M. J., & Chilingarian, I. 2015, *ApJ*, 806, 122